**Operasi File C++ – Menulis dan Membaca File**

By [Rachmat Santoso](https://www.blogger.com/profile/12595269060403524756" \o "author profile)— Monday, 4 April 2016 — [17 Comments](http://www.nblognlife.com/2016/04/operasi-file-c-menulis-dan-membaca-file.html#comments)— [Struktur Data](http://www.nblognlife.com/search/label/Struktur%20Data)

|  |
| --- |
| [https://4.bp.blogspot.com/-s6Pr47jMsXo/WOW0dFHPULI/AAAAAAAABBs/HCySn3VnER0urPCElU_Od0OIZzPTzV1CQCLcB/s1600/nblognlife.com%2B-%2BOperasi%2BFile%2BC%252B%252B%2B%25E2%2580%2593%2BMenulis%2Bdan%2BMembaca%2BFile.jpg](https://4.bp.blogspot.com/-s6Pr47jMsXo/WOW0dFHPULI/AAAAAAAABBs/HCySn3VnER0urPCElU_Od0OIZzPTzV1CQCLcB/s1600/nblognlife.com+-+Operasi+File+C%2B%2B+%E2%80%93+Menulis+dan+Membaca+File.jpg) |
| Contoh Program Operasi File C++ – Menulis dan Membaca File |

     File di sini dapat dikatakan sebagai penyimpanan data eksternal yang bersifat permanen. Operasi-operasi terhadap file berkaitan dengan input dan juga output serta hal lain seperti mengecek keberadaan suatu file, ukuran file, dan lain-lain. Contoh program ini akan menggunakan header “fstream”.

-       ofstream untuk menulis file

-       ifstream untuk membaca file

-       fstream untuk keduanya (menulis dan membaca)

**MENULIS FILE**

Syntax:

/\*

       Operasi File 1 (Menulis File)

       VS12 Express

       by R.S.

\*/

#include <iostream>

#include <conio.h>

#include <fstream>

using namespace std;

int main()

{

       //stream untuk menulis file

       ofstream myfile;

       //membuka file,

       //jika file tidak ditemukan maka file akan otomatis dibuat

       myfile.open("TEST.txt", ios::app);

       cout<<"OPERASI FILE 1"<<endl;

       cout<<"--------------"<<endl;

       //fail() -> untuk memeriksa suatu kesalahan pada operasi file

       if(!myfile.fail())

       {

              //menulis ke dalam file

              myfile<<"Belajar OPERASI FILE"<<endl;

              myfile.close(); //menutup file

              cout<<"Text telah ditulis ke dalam File"<<endl;

       }else{

              cout<<"File tidak ditemukan"<<endl;

       }

       \_getche();

       return EXIT\_SUCCESS;

}

Hasil

OPERASI FILE 1  
--------------  
Text telah ditulis ke dalam File

TEST.txt

|  |
| --- |
| [https://4.bp.blogspot.com/-uaFFD8dldJ4/VwIVNtVMaxI/AAAAAAAAAps/AOZe1u3ZqQcyzutksdfWLEk05FKzca1aA/s640/operasi%2Bfile%2BC%252B%252B%2Bmenulis%2B2.PNG](https://4.bp.blogspot.com/-uaFFD8dldJ4/VwIVNtVMaxI/AAAAAAAAAps/AOZe1u3ZqQcyzutksdfWLEk05FKzca1aA/s1600/operasi+file+C%2B%2B+menulis+2.PNG) |
| Operasi File C++ - File TEST.txt yang Berhasil Diciptakan |

Isi dari file TEST.txt

|  |
| --- |
| [https://4.bp.blogspot.com/-O0cyyzVonkE/VwIVtLn2PMI/AAAAAAAAAp0/sXD54B2M1MMFF33Z39AOGO_dERpyvtIHw/s640/operasi%2Bfile%2BC%252B%252B%2Bmenulis%2B3.PNG](https://4.bp.blogspot.com/-O0cyyzVonkE/VwIVtLn2PMI/AAAAAAAAAp0/sXD54B2M1MMFF33Z39AOGO_dERpyvtIHw/s1600/operasi+file+C%2B%2B+menulis+3.PNG) |
| Operasi File C++ - Isi dari file TEST.txt |

**MEMBACA FILE**

(Pada contoh ini kita akan membaca isi dari file dan menampilkannya ke layar.)

Syntax:

/\*

       Operasi File 2 (Membaca File)

       VS12 Express

       by R.S.

\*/

#include <iostream>

#include <conio.h>

#include <fstream>

#include <stdlib.h>

using namespace std;

int main()

{

       //stream untuk membaca file

       ifstream myfile;

       char sv\_text;

       //membuka file yang telah ada

       myfile.open("TEST.txt");

       cout<<"OPERASI FILE 2"<<endl;

       cout<<"--------------"<<endl;

       //fail() -> untuk memeriksa suatu pada kesalahan operasi file

       if(!myfile.fail())

       {

              cout<<"Isi dari File -> ";

              //ulang selama program belum mencapai akhir konten dari file

              while (!myfile.eof())

              {

                     myfile.get(sv\_text);

                     cout<<sv\_text;

              }

              myfile.close(); //menutup file

       }else{

              cout<<"File tidak ditemukan"<<endl;

       }

       \_getche();

       return EXIT\_SUCCESS;

}

Hasil

OPERASI FILE 2

--------------

Isi dari File -> Belajar OPERASI FILE

**Keterangan :**

|  |  |
| --- | --- |
| ios::in | Membuka file untuk operasi input. (untuk operasi baca *ifstream*) |
| ios::out | Membuka file untuk operasi output. (untuk operasi baca *ofstream*) |
| ios::binary | Membuka file untuk operasi baca dan tulis dalam biner. |
| ios::ate | Mengatur posisi awal pointer file pada akhir file. Jika posisi pointer tidak diatur/ditetapkan, posisi awalnya adalah  di awal file. |
| ios::app | Untuk menambahkan data pada suatu file. Data ditambahkan di akhir file, sehingga data sebelumnya pada file yang sama tidak hilang.  Jika file yang dimaksud tidak ditemukan maka program akan membuat file baru. |
| ios::trunc | Jika file digunakan untuk operasi output dan file telah ada sebelumnya, content sebelumnya akan dihapus dan digantikan dengan yang baru. |
|  |  |
|  | |
| bad() | Mengembalikan nilai TRUE jika operasi baca dan tulis mengalami kegagalan, yang disebabkan oleh kesalahan baca/tulis (Read/writing error).  Misalkan, kita mencoba untuk menulis ke dalam file yang tidak dibuka untuk menulis. |
| fail() | Mengembalikan nilai TRUE jika operasi baca dan tulis mengalami kegagalan, yang disebabkan oleh kesalahan logika (Logical error) atau kesalahan baca/tulis (Read/writing error). |
| eof() | Mengembalikan nilai TRUE jika file yang dibuka untuk dibaca telah mencapai akhir konten dari file. |
| good() | Mengembalikan nilai TRUE jika pada operasi file tidak terjadi kesalahan apapun. |

**Tambahan:**  
***ifstream*** secara otomatis mengeset flag ***ios::in***. Begitu juga***ofstream*** secara otomatis mengeset flag ***ios::out***. Sehingga kita tidak perlu secara manual mencantumkan kedua flag tersebut.   
Namun ketika kita menggunakan***fstream*** kita harus mencantumkan secara manual (menulisnya secara jelas/eksplisit) ***ios::in*** atau ***ios:out***.

----------------------------------------

**OPERASI BERBASIS OBJEK**

**MENULIS**

Suatu objek dapat ditulis ke dalam suatu file dengan menggunakan fungsi anggota **write()**.

Bentuk penulisan:

*objek.write((char \*)&objek, sizeof(objek));*

**MEMBACA**

Suatu objek  yang terdapat pada suatu file dapat dibaca dengan menggunakan fungsi anggota **read()**.

Bentuk penulisan:

*objek.read((char \*)&objek, sizeof(objek));*

Keterangan :

*objek*: objek untuk operasi file (objek dari stream)

*objek*: objek dari suatu kelas atau struktur

Berikut adalah contoh program yang menggunakan operasi berbasis objek untuk menulis dan membaca file :

/\*

      Operasi Berbasis Objek

      Microsoft Visual Studio Express 2012

      [RS]

\*/

#include <iostream>

#include <fstream>

#include <conio.h>

using namespace std;

//deklarasi struct bernama Input

struct Input

{

      float data;

}obj1, obj2; //objek dari struct Input

//deklarasi struct untuk Hasil

struct Hasil

{

      float jumlah,kurang,kali,bagi;

}result; //objek dari struct Hasil

//Catatan : Objek digunakan untuk mengakses anggota struct

//fungsi untuk melakukan input dan menyimpan ke dalam FILE

void input(Input obj1, Input obj2)

{

      /\*- ofstream adalah stream class untuk menulis pada FILE

        - dataInput1 dan dataInput2 merupakan objek dari ofstream

        - data yang diinput disimpan dalam "Input1.txt" dan "Input2.txt"

      \*/

      ofstream dataInput1("Input1.txt");

      ofstream dataInput2("Input2.txt");

      cout<<"\nMasukkan bilangan pertama : "; cin>>obj1.data;

      cout<<"Masukkan bilangan kedua : "; cin>>obj2.data;

      //Menyimpan ke dalam FILE

      dataInput1.write((char \*)&obj1, sizeof(obj1));

      dataInput2.write((char \*)&obj2, sizeof(obj2));

      //menutup FILE

      dataInput1.close();

      dataInput2.close();

}

//fungsi untuk melakukan penghitungan

void hitung(Input obj1, Input obj2)

{

      //pendeklarasian variabel

      float bil1, bil2;

      Hasil result;

      ofstream dataHasil("Hasil.txt");

      //ifstream : stream class untuk membaca FILE(Input1.txt)

      ifstream bukaInput1("Input1.txt");

      bukaInput1.read((char \*)&obj1, sizeof(obj1));

      //operasi penugasan, bil1 = nilai “data” pada obj1

      bil1 = obj1.data;

      ifstream bukaInput2("Input2.txt");

      bukaInput2.read((char \*)&obj2, sizeof(obj2));

      //operasi penugasan, bil1 = nilai “data” pada obj2

      bil2 = obj2.data;

      //hasilnya disimpan dalam objek struct -> "result"

      result.jumlah = bil1 + bil2;

      result.kurang = bil1 - bil2;

      result.kali = bil1 \* bil2;

      result.bagi = bil1 / bil2;

      //menulis hasil berupa objek ke dalam FILE

      dataHasil.write((char \*)&result, sizeof(result));

      //menutup FILE

      bukaInput1.close();

      bukaInput2.close();

      dataHasil.close();

}

//fungsi untuk menampilkan hasil

void hasil(Hasil result)

{

      ifstream bukaHasil("Hasil.txt");

      /\*membaca data berupa objek yang ada dalam FILE "Hasil.txt")\*/

      bukaHasil.read((char \*)&result, sizeof(result));

      //menampilkan hasil pengaksesan FILE

      cout<<"\nHASIL : "<<endl;

      cout<<"-------"<<endl;

      cout<<"Hasil penjumlahan = "<<result.jumlah<<endl;

      cout<<"Hasil pengurangan = "<<result.kurang<<endl;

      cout<<"Hasil perkalian   = "<<result.kali<<endl;

      cout<<"Hasil pembagian   = "<<result.bagi<<endl;

      //menutup FILE

      bukaHasil.close();

}

//fungsi main

int main()

{

      cout<<"OPERASI ARITMATIKA ANTARA DUA FILE"<<endl;

      cout<<"=================================="<<endl;

      //pemanggilan fungsi

      input(obj1, obj2);

      hitung(obj1, obj2);

      hasil(result);

      \_getche();

      return 0;

}

**Output**

OPERASI ARITMATIKA ANTARA DUA FILE

==================================

Masukkan bilangan pertama : 10

Masukkan bilangan kedua : 4

HASIL :

-------

Hasil penjumlahan = 14

Hasil pengurangan = 6

Hasil perkalian   = 40

Hasil pembagian   = 2.5